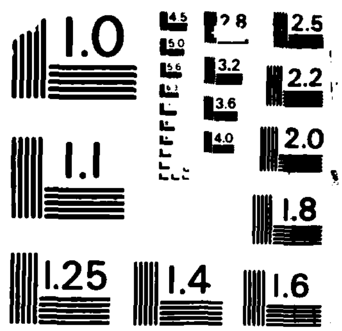
AD-A190 144 DESIGN FOR DATA EXTRACTION FROM DIGITIZED CONTOURS(U) 1/1
SCIENCE APPLICATIONS INC MCLEAN VA OCEAN ACOUSTICS DIV
R G STIEGLITZ DEC 81 SAI-82-656-WA N00014-79-C-0973

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD-A190 144

DTIC FILE COPY

DESIGN FOR DATA EXTRACTION
FROM DIGITIZED CONTOURS

SAI-82-656-WA

DTIC
ELECTE
FEB 1 0 1988
S D
D

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

SCIENCE APPLICATIONS, INC.
Post Office Box 1303, 1710 Goodridge Drive, McLean, Virginia 22102, (703) 821-4300

88 2 05 049

DESIGN FOR DATA EXTRACTION FROM DIGITIZED CONTOURS

SAI-82-656-WA

SCIENCE APPLICATIONS, INC.

Post Office Box 1303, 1710 Goodridge Drive, McLean, Virginia 22102, (703) 821-4300

DESIGN FOR DATA EXTRACTION FROM DIGITIZED CONTOURS

SAI-82-656-WA

December 1981

Prepared by:
Robert G. Stieglitz
Ocean Acoustics Division

Prepared for:
CDR M. A. McCallister
Code 522
Naval Ocean Research and Development Activity
NSTL Station
Bay St. Louis, MS 39520

Contract N00014-79-C-0973

Science Applications, Inc.

1710 Goodridge Drive
P.O. Box 1303
McLean, VA 22102
(703) 821-4300

TABLE OF CONTENTS

Section 1
INTRODUCTION

1.1 THE BASIC PROBLEM

The problem addressed by this exercise is the extraction of a sequence of points ("ranges") along specified tracks (generally radials emanating from a given point) which intersect digitized contour lines (of e.g., bathymetry), along with the associated contour values. The points are then sorted to provide the contour values as a function of range along the sequence of radials.

The digitized contours and radials are specified as a sequence of (x,y) [longitude, latitude] pairs. A Cartesian geometry is assumed. Since distances are computed in the standard Cartesian way, it is totally irrelevant whether the radials and contours are specified in terms of an absolute origin or in terms of a local relative origin. This latter feature of course has advantages when the absolute coordinates are sensitive. In addition, the method makes no assumption about whether the contours are closed or open in the "universe" under consideration.

Section 2
MATHEMATICAL METHOD

The radials under consideration in this problem are considered to be simple one-component line segments. The digitized contours are considered to be a sequence of line segments, which may or may not be closed. The mathematical solution of the intersection of a line (radial) with a sequence of segments (contour) depends on an interesting parameterization of a line segment. Given a line segment defined by its endpoints (x_1 and y_1) and (x_2 and y_2), we choose to express the equation of the line in the following parametric form.

$$x = x_1 + t_1 * (x_2 - x_1)$$

$$y = y_1 + t_1 * (y_2 - y_1) \quad \text{... the first line segment}$$

When $t_1 = 0$, x and y reduce to (x_1 and y_1) and when $t_1 = 1$, x and y reduce to (x_2, y_2). For $0 \le t_1 \le 1$, we have the point (x,y) lying somewhere on the line segment defined by (x_1, y_1) and (x_2, y_2). For $t_1 < 0$ or $t_1 > 1$, the point (x,y) is somewhere on the extension of the original segment but external to it.

We will now parameterize a second line segment, defined by its endpoints (x_3, y_3) and (x_4, y_4) in a similar fashion, using the parameter t_2.

$$x = x_3 + t_2 * (x_4 - x_3)$$

$$y = y_3 + t_2 * (y_4 - y_3) \quad \text{... the second line segment}$$

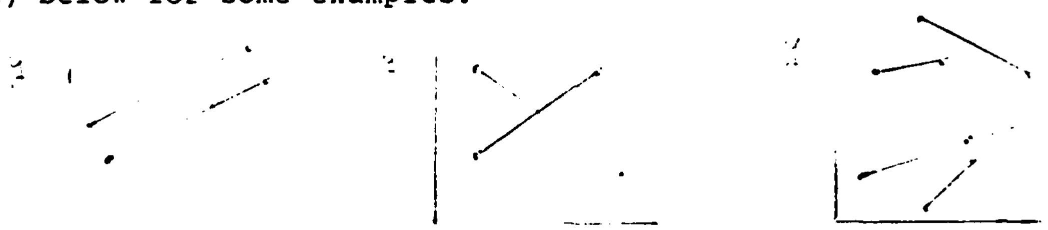
Now equate the two values of x and the two values of y and solve for the values of t_1 and t_2. We have the following

$$t_1 = \frac{-(x_3-x_1)(y_4-y_3) + (y_3-y_1)(x_4-x_3)}{-(x_2-x_1)(y_4-y_3) + (y_2-y_1)(x_4-x_3)}$$

and

$$t_2 = \frac{(x_2-x_1)(y_3-y_1) - (y_2-y_1)(x_3-x_1)}{-(x_2-x_1)(y_4-y_3) + (y_2-y_1)(x_4-x_3)}$$

Note that the denominators are equal. If the common denominator is zero, then the two line segments are parallel and do not intersect. See case a) below. If both t_1 and t_2 are greater than zero and less than 1, then the line segments intersect. See case b) below. If either t_1 or t_2 are less than zero or greater than 1, then the two segments do not intersect along the extent of their definition. See case c) below for some examples.



(a) (b) (c)

We simply apply this test, checking a given radial in turn against each line segment making up each contour. Remember that simple Cartesian geometry is used. Also note that there is no requirement that the contours be closed.

Section 3
DATA BASE FORMAT

The file of digitized contours will necessarily be supplied from an existing data base. In the present computer implementation, the following format is assumed. There are four records in the formatted file for each contour. Their content is as follows.

Record 1.	FORMAT(1X,I3)	IVAL ...	The contour value
Record 2.	FORMAT(A10)	X ...	A dummy value
Record 3.	FORMAT(4X,I4)	NPTS ...	The number of points in the contour
Record 4.	FORMAT(2F10.2)	(ALON(I), ALAT(I),I=1,NPTS) ...	The LONGITUDE-LATITUDE pairs making up the contour

When changing or modifying this code for another data base, it will be necessary to modify the four statements in the present code. The changes may include all or some of the following.

a. Modification of format to accept binary input.

b. A change in the order of the parameters.

c. The number of READ statements necessary to accomplish the task.

The input section is thus seen to contain no serious restrictions on the format of the data base. Note though that the contours are presented totally one after the other, each is complete in itself. No other order of contour sequencing is expected or required by the code.

Section 4

COMPUTER IMPLEMENTATION

The code as presented in the appendix is straightforward. The following will serve as a simple narrative of the computation.

a. (Fortran label 10) Get the contour from a disc file. As mentioned above, this segment is likely to change in another case. Skip to the end of the computation section (Fortran label 80) if there are no more contours to be considered.

b. Determine contour limits in spatial terms.

c. DO-loop on radials ... DO 50 NR = 1, NRADS.

d. Determine physical limits of the given radial; skip consideration of this radial if there is no possible intersection.

e. DO-loop on contour segments ... DO 40 I = 1, NMI, where NMI is defined to be one less than the number of points in the contour under consideration.

f. Check for possible intersection of given radial with current segment of contour. Store relevant results if so.

g. End DO-loop on points in this contour (label 40) and on the number of radials under consideration in this problem (here, fixed at five).

h. (Fortran label 80) Sort intersections for each radial into increasing order of range and then return.

VARIABLE Dictionary ...

ALON(1000)	longitude values for a given contour
ALAT(1000)	latitude values for a given contour
NPTS	number of points in current contour
X1S(5), Y1S(5)	starting longitude and latitude values for each of five radials
X2S(5), Y2S(5)	final point (longitude and latitude) for each of five contours
SAVRAD(300,5)	for each of 300 total radial-contour intersection in the problem the following:

longitude value	(-,1)	
latitude value	(-,2)	
range (nm)	(-,3)	
radial ordinal	(-,4)	(in the range 1-5)
contour value	(-,5)	

Section 5
POSSIBLE EXTENSIONS TO THE CODE

There are always many possible extensions or modifications to a computer code as new requirements arise. The following is a list of some of the more likely ones for this code.

a. The number and dimension of the radials permitted should be input to the code in its role as either a main program or as a subroutine.

b. The limit of 300 total radial-contour intersections is likely to prove restrictive.

c. (In conjunction with b) immediately above) A cutoff on the maximum range of a radial-contour intersection may be desirable in some cases.

d. The code segment towards the beginning governing the specific form of the digitized contour database will almost certainly be changed.

APPENDIX

```
      PROGRAM CWS(OUTPUT,TAPE6=OUTPUT,TAPE1,TAPE2)
C
      DIMENSION ALAT(1000),ALON(1000)
      DIMENSION X1S(5),X2S(5),Y1S(5),Y2S(5)
      DIMENSION SAVRAD(300,5)
      DATA X1S/5*-30.0/
      DATA Y1S/5*35.0/
      DATA X2S/-10.0,-40.0,-60.0,-55.0,-10.0/
      DATA Y2S/55.0,60.0,40.0,10.0,10.0/
C
C*********************************************************************
C          ALON ... LONGITUDE ... POSITIVE EAST, NEGATIVE WEST ( X COORDINATE
C          ALAT ... LATITUDE ... POSITIVE NORTH, NEGATIVE SOUTH ( Y COORD)
C
C          X1S,Y1S X2S,Y2S ARE THE START POINT AND STOP POINT OF FIVE
C          ARBITRARY RADIALS CENTERED AT 30W, 35N
C
C          RESULTS ARE IN SAVRAD(I,J), WHERE I IS SIMPLE INTERSECTION
C          ORDINAL AND J REFERS TO THE FOLLOWING ...
C               J = 1 , 2 , 3 , 4 , 5 ... INTERSECTING LONGITUDE (X)
C               AND LATITUDE (Y), RANGE (NM) FROM RADIAL ORIGIN, RADIAL
C               INDEX AND VALUE OF PARENT CONTOUR
C
C          A RECTANGULAR CARTESIAN COORDINATE SYSTEM IS ASSUMED
C
C*********************************************************************
      REWIND 1
      KINT = 0
      NCONT = 0
      NINTMX = 300
      NRADS = 5
      XMAX = 9.0
      YMAX = 9.0
C
   91 FORMAT(1X,I3)
   92 FORMAT(10A10)
   93 FORMAT(4X,I4)
   94 FORMAT(2F10.2)
   95 FORMAT(//,3I5)
   96 FORMAT(5F15.6)
C
C
C**********        START THE CONTOUR LOOP.
   10 CONTINUE
C*********************************************************************
C          THE FOLLOWING FORMAT FOR THE DIGITIZED CONTOURS IS OF COURSE
C          SPECIFIC TO THIS IMPLEMENTATION BUT NOT AT ALL CRUCIAL TO THE
C          OVERALL ALGORITHM.
C          NOTE ALSO THAT WE TERMINATE THE DEMONSTRATION AT 75 CONTOURS
C*********************************************************************
C          IVAL IS CONTOUR VALUE
      READ(1,91) IVAL
      IF(EOF(1).NE.0.0) GO TO 80
      NCONT = NCONT + 1
      IF(NCONT.GT.75) GO TO 80
C          X IS A DUMMY VARIABLE
      READ(1,92) X
      READ(1,93) NPTS
      READ(1,94) (ALON(I),ALAT(I),I=1,NPTS)
      XMX = YMX = -1.0E20
      XMN = YMN = 1.0E20
```

```
      DO 20 I = 1 , NPTS
C**********     DETERMINE CONTOUR MAX AND MIN
      XMX = AMAX1(XMX,ALON(I))
      XMN = AMIN1(XMN,ALON(I))
      YMX = AMAX1(YMX,ALAT(I))
      YMN = AMIN1(YMN,ALAT(I))
   20 CONTINUE
C**********     LOOP ON RADIALS
      DO 50 NR = 1 , NRADS
C**********     GET RADIAL LIMITS
      X1 = X1S(NR)
      Y1 = Y1S(NR)
      X2 = X2S(NR)
      Y2 = Y2S(NR)
C**********     RADIAL X AND Y MAX AND MIN
      RXX = AMAX1(X1,X2)
      RXN = AMIN1(X1,X2)
      RYX = AMAX1(Y1,Y2)
      RYN = AMIN1(Y1,Y2)
C**********     IS THE CURRENT RADIAL RELEVANT TO THE CURRENT CONTOUR
      IF(RXX.LT.XMN) GO TO 50
      IF(RXN.GT.XMX) GO TO 50
      IF(RYX.LT.YMN) GO TO 50
      IF(RYN.GT.YMX) GO TO 50
C        LOOP OVER CONTOUR SEGMENTS
C        RADIAL RUNS FROM (X1,Y1) TO (X2,Y2)
C        CONTOUR SEGMENTS RUN FROM (X3,Y3) TO (X4,Y4)
      NM1 = NPTS - 1
      DO 40 I = 1 , NM1
      X3 = ALON(I)
      X4 = ALON(I+1)
      Y3 = ALAT(I)
      Y4 = ALAT(I+1)
C*************************************************************************
C         APPLY ALGORITHM TO SEE IF SEGMENTS ARE PARALLEL,ONLY APPARENTLY
C         INTERSECT OR ACTUALLY INTERSECT WITHIN THEIR EXTENT
C*************************************************************************
      DEN = -(X2-X1)*(Y4-Y3) + (Y2-Y1)*(X4-X3)
      IF(DEN.EQ.0.0) GO TO 40
      ANUMA = -(X3-X1)*(Y4-Y3) + (Y3-Y1)*(X4-X3)
      TALPHA = ANUMA / DEN
      IF(TALPHA.LT.0.0) GO TO 40
      IF(TALPHA.GT.1.0) GO TO 40
      ANUMB = (X2-X1)*(Y3-Y1) - (Y2-Y1)*(X3-X1)
      TBETA = ANUMB / DEN
      IF(TBETA.LT.0.0) GO TO 40
      IF(TBETA.GT.1.0) GO TO 40
C        FOUND ONE
C        (X,Y) IS (LONGITUDE,LATITUDE) OF INTERSECTION
      X = (X2-X1)*TALPHA + X1
      Y = (Y2-Y1)*TALPHA + Y1
      NINT = NINT + 1
      IF(NINT.GT.NINTMX) GO TO 60
      SAVRAD(NINT,1) = X
      SAVRAD(NINT,2) = Y
      SAVRAD(NINT,3) = 60.0 * SQRT( (X-X1)**2 + (Y-Y1)**2 )
      SAVRAD(NINT,4) = FLOAT(NR)
```

```
      SAVRAD(NINT.5) = FLOAT(IVAL)
   40 CONTINUE
   50 CONTINUE
      GO TO 10
C
   80 CONTINUE
C        SORT INTERSECTIONS INTO ASCENDING RANGE
      NM1 = NINT - 1
      DO 76 N1 = 1 . NM1
      N2 = N1 + 1
      DO 75 N3 = N2 . NINT
      IF(SAVRAD(N1.3) .LE.SAVRAD(N3.3) ) GO TO 75
      DO 74 N4 = 1 . 5
      TEMP = SAVRAD(N1.N4)
      SAVRAD(N1.N4) = SAVRAD(N3.N4)
      SAVRAD(N3.N4) = TEMP
   74 CONTINUE
   75 CONTINUE
   76 CONTINUE
      WRITE(6.81)
   81 FORMAT(*1 SUMMARY OF RADIAL-CONTOUR INTERSECTIONS*./)
      DO 88 NR = 1 . NRADS
      WRITE(6.82)
   82 FORMAT(/)
      DO 86 I = 1 . NINT
      IF(FLOAT(NR) .NE. SAVRAD(I.4)) GO TO 86
      WRITE(6.83) I.(SAVRAD(I.J).J=1.5)
   83 FORMAT(I5.3F10.2.2F5.0)
   86 CONTINUE
   88 CONTINUE
      WRITE(6.82)
      WRITE(6.82)
      END FILE 6
      STOP
C
      END
?
```

END
DATE
FILMED
APRIL
1988
DTIC